An Authentication Protocol for Mobile Devices Using Hyperelliptic Curve Cryptography

Mrs. S. Prasanna Ganesan
Assistant Professor in Computer Science, Dr. GRD College of Science, Coimbatore -14
Email: prasannaraog@yahoo.co.in

***Abstract* — At present, most of e-commerce applications are developed using asymmetric cryptography to guarantee the authentication of the involved parties. On the other hand, a growing demand for mobile devices has geared a shift towards mobile e-commerce applications. This paper highlights that the existing authentication protocols, based on RSA asymmetric cryptography, are not appropriate for such devices due to their limitations in computing power, memory capacity, key sizes and cryptographic support. Therefore, an efficient protocol for resource constrained platforms that achieve a level of security similar to the one achieved by the protocols in use today is designed and implemented. This protocol is based solely on Hyperelliptic curve asymmetric cryptography and the results prove that the performance achieved is good compared to RSA.**

***Index Terms:*Hyperelliptic curve cryptography, RSA, SSL, TLS, SET, PDA, J2ME, key generation, sign generation, sign verification.**

I. INTRODUCTION

The remarkable growth of communication technologies and the extensive use of the internet have contributed to the development and budding of m-commerce. Conversely, we have seen a growing demand for mobile devices. This seeks for smaller, cheaper and faster platforms has guided to the appearance of PDAs, Cellular phones and pagers. Therefore, even though the PC platform has been the foremost target for client applications, we are able to expect a migration of e-commerce applications from the conventional desktop to these mobile devices. For example, one might think of buying/selling products through a mobile phone or browsing pay-per-view news on PDA, while waiting on the bus stop.

However, being the internet an open and insecure network, some anxiety has been raised in transmitting sensitive information. The solution lies in using cryptography and secures authentication protocols that guarantee the confidentiality, authentication and integrity of communications [4]. Such protocols, like SSL [11] and SET [10], already exist and are widely used in current e-commerce applications. Most of them are based in RSA public key cryptography. A protocol is developed which is based exclusively on hyperelliptic curve cryptography (HECC), an asymmetric cryptography that performs well in resource constrained platforms and maintain the high security level that one can achieve with the protocols in use today.

The paper is organized as follows. Section 2 describes the main aspect of the protocol. In section 3, the implementation of the protocol in J2ME wireless toolkit is provided. Finally, Section 4 presents the conclusions.

II. ARCHITECTURE

The authentication protocol must be able to create a secure channel between two principals on top of an insecure network, like the internet. It's not difficult to eavesdrop a line or to compromise a router and be able to listen / alter all messages in transit [1]. In order to prevent this, the protocol must ensure the mutual authentication of both parties and the confidentiality and integrity of all the data transmitted through it. Such protocols already exist and have gone through deep analysis, like SSL [11] and TLS [12]. However, they rely heavily on RSA asymmetric cryptography, which causes some concern about their performance on resource constrained small devices. In fact, some performance measurement is done for cryptographic functions on one of these devices, the PalmIII from 3Com [9]:

Table 1: Performance measurement for RSA cryptographic algorithm

Algorithm	Key Length (bits)	Time taken in PalmIII
RSA key generation	512	165000 milliseconds
RSA sign generation	512	5000 milliseconds
RSA sign verification	512	640 milliseconds

As we can see, generating RSA keys on the PalmIII Pilot is prohibitively expensive and time consuming one. Moreover, an RSA signature generation is also very slow and RSA itself is vulnerable [8], which makes a protocol like SSL to become unusable. Conversely, the implementation of HECC asymmetric-key on mobile perform well and provide more secure even if we consider the key length of 80 bit compared to RSA's 1024 bit [2] [3] [5] , which lead to a simple conclusion: the protocol must be based solely on HECC asymmetric-key cryptography. The following figure 1 depicts the authentication protocol over the internet by using HECC asymmetric key cryptographic algorithm.

DOI: 01.IJNS.02.01.40

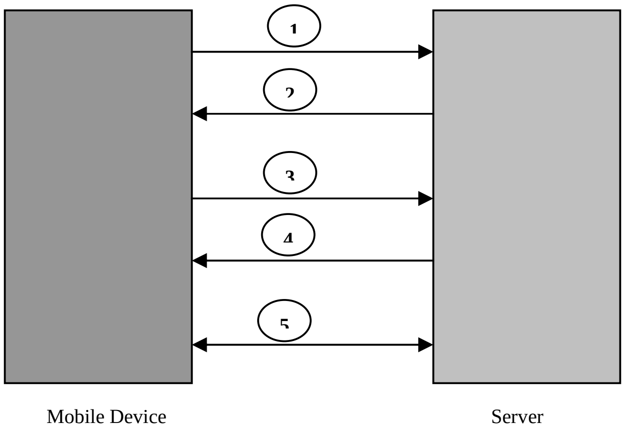


Figure 1: The asymmetric authentication protocol over the internet using HECC

The idea behind this protocol is simple: in step 1, the mobile starts the protocol by sending its ID (e.g. Serial Number) to the server. In step 2, the server stores the mobile's ID for authentication purpose and generates mobile's private key and public key using the hyperelliptic curve over prime field of genus 2 and its divisor. These keys (private and public key of the mobile) along with the public key of the server are sent to the mobile. Notice that the keys travel from the server into the mobile through a secure channel. To send the key to the respective destination, one can even adopt Diffie-Hellman key exchange algorithm.

In step 3, the mobile generates a challenge and sends it along with its ID to the server, encrypted with a combination of the server's public key and the mobile's private key. The server decrypts the message with mobile's public key and its private key and verifies if this ID matches the ID sent in step 1. This authenticates the client.

In step 4, the server sends the challenge received in the previous step plus one and a randomly generated session key and encrypted with a combination of mobile's public key and server's private key. The mobile then decrypts this message with server's public key and its private key and verifies the challenge. If it matches the one that was sent in step 3, then the mobile can trust that it's indeed talking to the right server. Both encryption and decryption process, specified in step 3 and 4 are done using hyperelliptic curve cryptographic technique.

From now on, in step 5, a secure channel has been created and all data is encrypted with a session key. Notice that a new key is setup for each message to prevent replay attacks.

III. Implementation

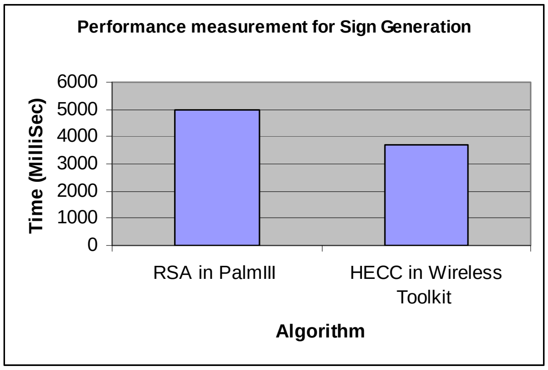
The proposed protocol using hyperelliptic curve over prime finite field of genus 2 with 80 bit was successfully implemented in a J2ME wireless tool kit 2.5.1. The details of the Sun Java Wireless Toolkit 2.5.1 can be had from [6] and the toolkit can be downloaded from [7]. With the growing diversity of mobile devices to which the protocol targeted for, its portability was a major concern since the beginning.

Therefore it was developed using the J2ME, whose features meet this requirement. To achieve the high security level required, the ElGamal based hyperelliptic curve cryptography and MD5 algorithms were used for the encryption, decryption and digest calculation of the messages exchanged in this protocol. Table 2 shows the performance measurement for hyperelliptic curve cryptographic algorithm over finite field Fp of genus 2 on J2ME wireless toolkit 2.5.1. Figure 2 shows the comparative analysis of performance measurement of both RSA and HECC.

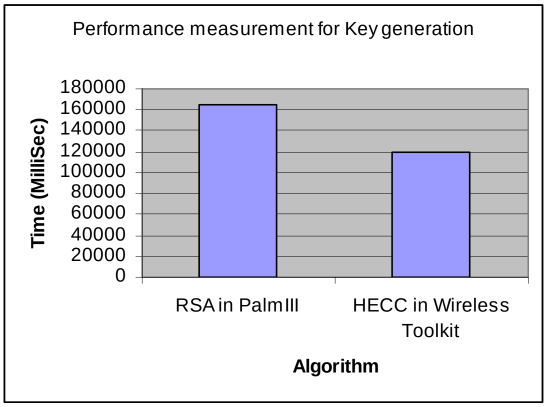
Table 2: Performance measurement for hyperelliptic curve cryptographic algorithm

Algorithm	Key Length (bits)	Time taken in Wireless Toolkit 2.5.1
HECC key generation	80	120000 milliseconds
HECC sign generation	80	3700 milliseconds
HECC sign verification	80	520 milliseconds

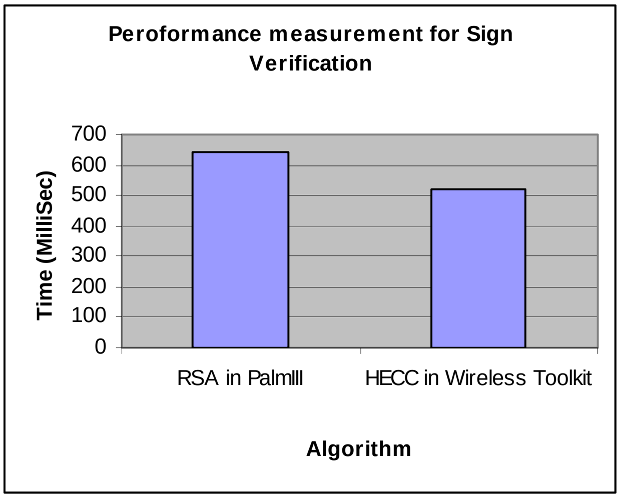
(Fig. 2a)



(Fig. 2b)



DOI: 01.IJNS.02.01.40



(Fig. 2c)

Figure 2: Comparative analysis of performance measurement of both RSA and HECC

IV. Conclusion

This work shows that it is possible to implement the authentication protocol using HECC in resource constrained mobile devices with reasonable performance compared to RSA. Protocols based on this HECC asymmetric cryptography can be directly used in such devices. This paper addressed the design of a protocol based on HECC asymmetric cryptography. Furthermore, an implementation for J2ME Wireless Tool Kit 2.5.1 is also described. Hope this work to be a big contribution to the development and widespread acceptance of m-commerce.

Acknowledgment

The author wish to sincerely thank the management of Dr. GRD College of science, Coimbatore for their constant encouragement and financial support rendered during the course of this research work.

REFERENCES

[1] CERT Advisory CA-95.01. - IP Spoofing Attacks and Hijacked Terminal Connections

[2] Fan X and Gong G, “Efficient Explicit Formulae for Genus 2 Hyperelliptic Curves over Prime Fields and Their Implementations”, Department of Electrical and Computer Engineering, University of Waterloo,Waterloo,Ontario N2L 3G1, CANADA; www.cacr.math.uwaterloo.ca/techreports/2007/ cacr2007-01.pdf

[3] Ganesan R and Dr. Vivekanandan K, ”Performance Analysis of Hyperelliptic Curve Cryptosystems over Finite Field F_p for Genus 2 and 4”, International Journal of Computer Science and Network Security (IJCSNS), Journal ISSN:1738-7906, Vol.8, No.12, December 2008, pp 415 – 418.

[4] Ganesan R, Gobi M and Dr. Janakiraman VS, “Implementation of MD5 Integrity Checking Mechanism for M-Commerce Transactions”, International Journal of Computer Science and Applications, Vol.1, No.3, December 2008, pp.194 -196, ISSN 0974-1003.

[5] Ganesan R, Gobi M and Dr. Vivekanandan, ”Elliptic and Hyperelliptic Curve Cryptography Over Finite Field F_p“, i-Manager’s Journal on Software Engineering, Vol.3, Issue No.2, October-December, 2008, pp 43-48, ISSN-0973-5151.

[6] http://java.sun.com/javame/reference/apis.jsp

[7] http://java.sun.com/products/sjwtoolkit/download-2_5_1.html

[8] Imad Khaled Salah, Abdullah Darwish and Saleh Oqeili, “Mathematical attacks on RSA cryptosystem”, Journal of Computer Science, August, 2006.

[9] Neil Daswany, Dan Boneh, Experimenting with Electronic Commerce on the PalmPilot, Stanford University, 1998.

[10] The SET Standard Specification; http://www.setco.org/set_specifications.html. 1999.

[11] The SSL Protocol Version 3.0. Netscape Communications, 1996.

[12] Transport Layer Security Working Group. The TLS Protocol (Internet-Draft). 1997.

DOI: 01.IJNS.02.01.40